数理解析研究所講究録
585巻 1986年 81-95

大規模回路シミュレーションに対する共役残差法の適用実験

（株）日立製作所中央研究所　山本富士男(Yamamoto Fujio)
〃　梅谷　征雄(Umetani Yukio)
〃　高橋　栄　(Takahashi Sakae)

０．はじめに

大規模非対称不規則疎行列の典型である、回路シミュレーションでの行列に対しては、直接系解法が常用されてきたが、対象回路規模の増大に伴い、必要メモリ容量と計算速度の両面でさらに高性能な解法が必要である。ここでは、一つの試みとして、共役残差法とＬＵ分解を結合した方式を構成し、現実のＶＬＳＩ回路シミュレーションに適用してその性能特性を評価した。

１．回路シミュレーションの概要

回路シミュレータは、デバイス特性と回路図情報を入力とし、回路動作波形を計算で求めるものであり、それが使われるフェーズは２ヶ所ある。（図１．１）　最初は回路設計段階であり、次はレイアウト設計段階である。後者では作成されたマスクパタンから直接回路動作を確認できるため、今後の大規模回路を精密に設計するうえで特に重要である。この場合、マスクパタンからはプログラムによって、寄生素子を含む回路図情報が自動的に復元されるため、回路シミュレーションの対象となる回路は極めて大規模になる傾向がある。回路シミュレータには、このような大規模回路に対しても、図１．２に示すような、シンクロスコープに表示されるものに近い精密な波形を出力することが求められる。

回路シミュレーションの計算内容を外部的に見ると、図１．３のように、デバイスモデル計算部と行列計算部の繰り返しで構成されており、行列解法に従来のＬＵ分解

法を用いた場合、大規模では圧倒的に行列計算時間が大きかった。これに対し、開発済のベクトル化ＬＵ分解方式[1)]を用いると、図中の棒グラフに示すように著しい効果があった。しかし、デバイスモデル計算は規模に比例して増加するのに対し、行列計算はこの方式でも、規模の１．３～１．９乗で増加する。また、メモリ量も同程度の割合で増加する点が大きな問題であり、今後の大規模化でさらに深刻化すると考えられる。したがって、行列演算の高性能化の研究開発は依然不可欠である。

２．回路行列の特性と解法

２．１　回路行列特性

回路行列の最大の特徴は図２．１に示すように、スパース率が非常に高く、非零要素の存在位置が非対称で不規則なことである。また、動的な特性として、行列の条件数は時としてかなり大きくなる。図２．２に示したものは、比較的条件の良い例である。

２．２　回路行列の解法

回路シミュレーションの方式は幾つかある（図２．３）が、ここでは現実の回路に対して、現時点では最も確実なMatrix-Based Methodをとる。したがって、これ以降、非対称不規則疎行列の解法について議論する。　解法として、上に述べた背景から、これまでの直接系解法に替えて、反復系解法を試みることにした。本報告では、図２．３に示した内で特に共役残差法を取り上げた。その動機および第３章で述べる方式を構成するに至った経過は次の通りである。

共役残差法適用の動機

（１）論理的に自然な解法であり，分かりやすい。
（２）ＬＵ分解に比べ，記憶容量，演算量／反復は少なそうである。
（３）スーパーコンピュータ上で使われ始めた。（分布定数系）

回路行列への適用上の問題点

（１）前処理無しＣＲ法，不完全ＬＵ分解（fill-in除く）付ＣＲ法
　　ではほとんど収束しない。

（２）分布定数系での，Meijerink-Gustafsson 流前処理[2]に相当するものが行列構造上，見つからない。

方　針

（１）収束しない場合は，完全ＬＵ分解で求解する。そのＬＵで後続の行列を前処理する。（過渡解析に対応する行列）
（２）残差の停留を早期に検出する。

３．完全ＬＵ分解付共役残差法の構成

上記のような理由から、図３．１に示す完全ＬＵ分解付共役残差法（ＣＬＵＣＲ法）を新たに構成した。この方法では、ＣＲ法が収束しない場合、完全ＬＵ分解を実行し求解する。この時の分解結果を保存しておき、次の行列ではこれを用いて前処理し、ＣＲ反復に入る。過渡解析の時間刻みは通常十分小さいことから、完全ＬＵ分解起動率はかなり低く押えられると考えた。一方、起動された場合は、ＣＲ法反復は無駄になるので、図に示したように、解の修正状況パラメータα、βによる残差停留の検出を行い、適切に不要な反復を打ち切る工夫を施した。

４．残差減少状況の考察

ここでは、９６元行列を有する１つの回路について、ＣＬＵＣＲ法の収束特性を調べてみる。図４．１は、この回路の過渡解析で解かれた多数の行列の内から、ランダムに選択したケースについて、残差減少の様子を示したものである。この図から、収束しない場合には、残差は早期に停留していることが多いと言える。一方、残差は反復の途中で一時的に停留することがある。したがって、残差自体を観測するだけでは不都合が生じる。そこで、第３章で述べたα、βの観測による残差停留検出を行うことにしたわけである。（図４．２）

次に、この回路について、α、βの変化を、高速に収束する場合（図４．３）、収束するが低速な場合（図４．４）、および収束しない場合（図４．５）について示す。図４．３では、αは１～１０という、かなり大きな動きを示し、βも１には近く

ない。図４．４では、αは１より小さく、βも１に接近しており、収束速度が鈍っている。途中、残差の疑似停留があるが、今回の停留検出法では、反復を続行すべきと判定された。これに対し、収束しない図４．５では、αが急速に減少し、βが１に極めて近づいた点で、残差停留と判定され、反復は打ち切られた。

５．大規模回路への適用性評価

本章では、３５～３６６８元行列を持つ現実の回路１０題にＣＬＵＣＲ法を適用した結果を検討する。

５．１　完全ＬＵ分解起動率と反復回数

１０題の内、波形変化の特に激しい１題を除き、　完全ＬＵ分解起動率０．０１～０．０７（図５．２），すなわち、１００回の行列演算でＬＵ分解の必要回数は１～７回、　ＣＲ法（共役残差法）反復平均８～４０回（図５．１）でいずれも相対残差10^{-12}の収束解を得た。完全ＬＵ分解起動率がこのように低いのは、過渡解析では一般に波形変化が滑らかな区間の割合が比較的大きいためと思われる。広範囲の規模の回路行列に対して、ＣＲ法に入ってからの収束性は、少数回の完全ＬＵ分解の効果によって、著しく向上したと判断される。なお、今回考案した残差停留検出法により、ＣＲ法の収束に至るまでの反復回数は、最大で元の個数回反復を許容する場合の約半数で済むことも確認できた。（図５．１）

５．２　浮動小数点演算回数の分析

以上で、ＣＬＵＣＲ法の収束特性が明らかになったので、次に演算量の分析を行なう。図５．３は完全ＬＵ分解、行列変換及び収束解変換など反復の外側にある前後処理、ＣＲ法反復１回の演算量を各々浮動小数点演算回数で示したものである。この図から、特に小規模回路では、ＣＲ法反復１回の演算量は、完全ＬＵ分解より多いことがわかる。しかし、この演算量の差は、規模の増大で完全ＬＵ分解演算量が急増するとともに縮まる傾向が明らかであり、すでにこれが大きく逆転している回路も見られる。

５．３　直接解法との比較

直接解法（ＬＵ分解法）との性能比較を浮動小数点演算回数で示したのが図５．４である。この図では、行列元数の増大にともなう演算量の増加傾向を推定するのが目的なので、ＣＬＵＣＲ法では収束性を正規化して示した。すなわち、完全ＬＵ分解起動率とＣＲ法平均反復回数に関して、図５．２と図５．１のデータから、やゝ楽観的なＣａｓｅ１と悲観的なＣａｓｅ２の２ケースを設定して示した。小規模の場合には、完全と不完全のＬＵ分解演算量に大きな差がないこと、及び図３．１②③④でのＣＲ法反復内外の行列乗算が意外に多いことにより、ＣＬＵＣＲ法は直接解法より演算量が多くなる。しかし、規模の増大とともに、直接解法での完全ＬＵ分解演算量が急速に増加し、ＣＬＵＣＲ法内行列乗算は相対的に少なくなる。このため、Ｃａｓｅ１では約６０００元以上の行列から次第に直接解法を凌駕し始めると推定される。

図５．５は主記憶容量の比較である。この図には、行列演算に必要な主記憶容量のみを示した。ＣＬＵＣＲ法では、完全ＬＵ分解を高速に実行するのに必要データは、上記のように、その使用頻度が極めて低いことから、高速外部記憶であるＳ－８１０拡張記憶を使うものとした。ＣＬＵＣＲ法の主メモリ量は、ベクトル化による直接解法（ＢＶＡ法[1]）に比べ非常に少なく、規模の増大に対する増加率もかなり低い。図中には、１００００元規模の回路行列に対する主メモリ量の推定値も示した。カッコ内はシミュレーション全体に必要なメモリ量の合計である。　これから、ＣＬＵＣＲ法では、拡張記憶が使えるなら、１００００元行列の回路シミュレーションを主記憶３３ＭＢで実行できると推定される。なお、拡張記憶使用によるＵＳＥ時間の増加は、次節で示すように僅かである。

５．４　１万元回路行列に対する推定性能

今後出現が予想される、１００００トランジスタ規模の回路に対する性能予測を表５．１に示した。この場合、回路行列は１万元、計算時間点数は８００点と仮定した。表中、ＥはＣＬＵＣＲ法で完全ＬＵ分解に通常のＤｉｓｋを用いた場合、Ｆは、Ｓ－８１０拡張記憶を用いた場合である。表中のＣＰＵ時間は多数の大規模回路の実測データから、ＵＳＥ時間は他プログラムでの実績などを参考にして、データ転送量

で換算して算出した。Ｄｉｓｋ使用ではやはりＵＳＥ時間の増加は耐え難いことがわかる。これに対し、拡張記憶使用の場合は、ＣＰＵ１．３時間にたいしてＵＳＥ１．７時間と推定され、十分実用範囲である。また、毎回完全ＬＵ分解が必要な直接解法では、主メモリ量を減らすため拡張記憶を用いると、ＵＳＥ時間はＥの場合と同じく１０時間程度と推定され、実用的でない。１万元行列に対しては、以上の通り、ＣＬＵＣＲ法は主記憶容量の面で大きな効果が期待できるが、ＣＰＵ時間の削減は僅かである。ＣＬＵＣＲ法が直接解法を圧倒的に凌駕するのは、もう１桁規模が大きい回路に対してであると考えられる。

５．５　他の反復解法との比較

第２章図２．３に示した反復解法の内、ＣＲ法以外に演算量の少なさの面で直接解法に勝る可能性を残していたのは、反復改良法[3]、Ｓouthwell緩和法[4]であった。これらに、やはり完全ＬＵ分解を結合した方式を構成し、ＣＬＵＣＲ法と比較した。（図５．６）　この図から、規模の増大にともなって、ＣＬＵＣＲ法以外は完全ＬＵ分解の必要回数が極めて多くなり、収束性は非常に悪化することが推測された。したがって、ＣＬＵＣＲ法を用いるべきと考える。

６．結　言

（１）回路行列解法として，共役残差法とＬＵ分解法を組み合わせた方式を構成した。

- 過渡解析における行列の，比較的滑らかな変化を利用して，間欠的にＬＵ分解を起動する。
- 反復回数を低減させるため，解の修正状況から残差の停留を早期に検出する。

（２）実回路１０題（35～3668元行列）に用した結果，いずれも相対残差 10^{-12} の収束解を得た。

- 完全ＬＵ分解起動率　：　０．０１ ～ ０．０７
- ＣＲ法平均反復回数　：　８ ～ ４０回

（３）収束性と演算量の分析から，約６０００元以上の行列で，直接解法（ＬＵ分解）の性能を上回ると推定した。

（４）大規模回路でも，完全ＬＵ分解起動率が上記のように低いので，高速ベクトル化ＬＵ分解の大量データをS-810拡張記憶装置などに置くことができる。

（５）回路動作特性とＣＲ法非収束との因果関係についての考察は今後の課題である。

謝辞

共役残差法などの反復解法の性能特性について御教示いただいた、図書館情報大学村田健郎教授に感謝の意を表す。

参考文献

1）F.Yamamoto, S.Takahashi: Vectorized LU Decomposition Algorithms for Large-Scale Circuit Simulation; IEEE Trans. on CAD Vol. CAD-4, No.3, 1985, pp. 232-239

2）村田・小国・唐木：スーパーコンピュータ　科学技術計算への適用；丸善、昭和６０年３月、ｐｐ．１３６－１５３

3）フォーサイス、モウラー（渋谷・田辺共訳）：線形計算の基礎；培風館、昭和４４年１１月、ｐｐ．５８－６４，１３７－１４２

4）赤坂隆：数値計算；応用数学講座第７巻、コロナ社、昭和６０年６月、ｐｐ．２４７－２６５

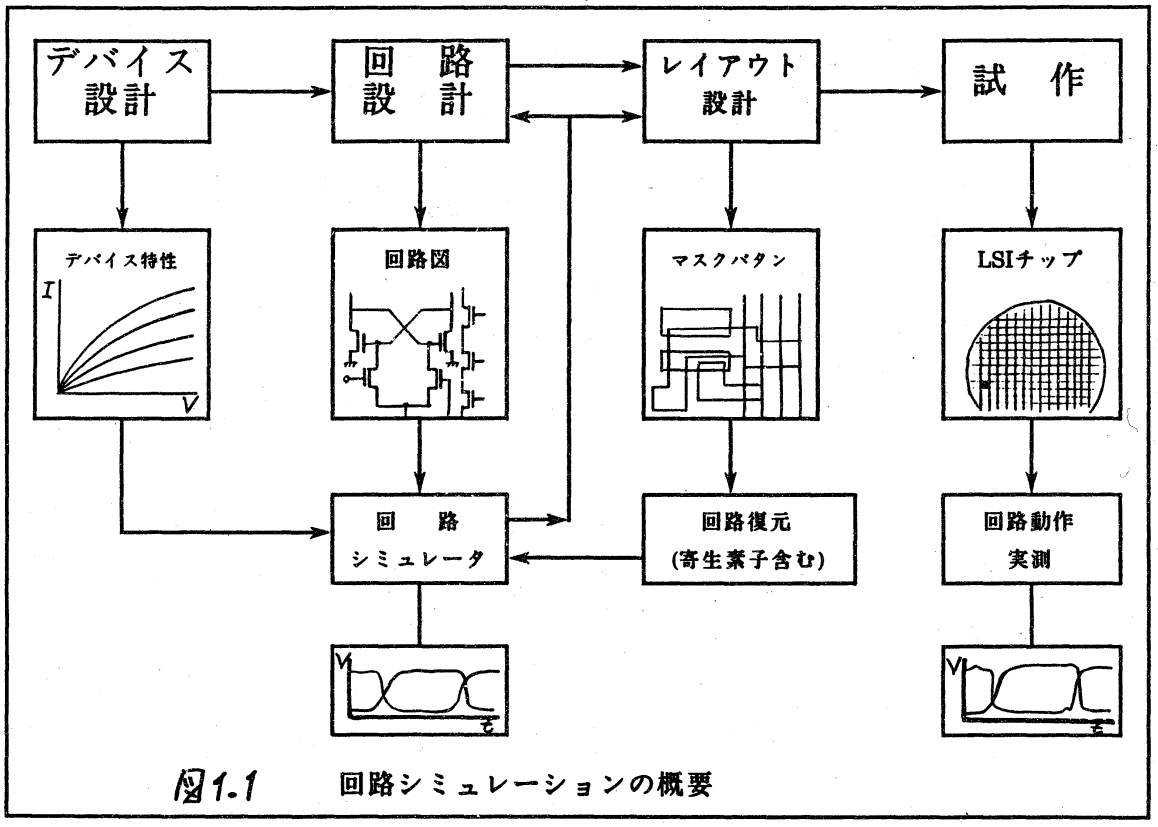
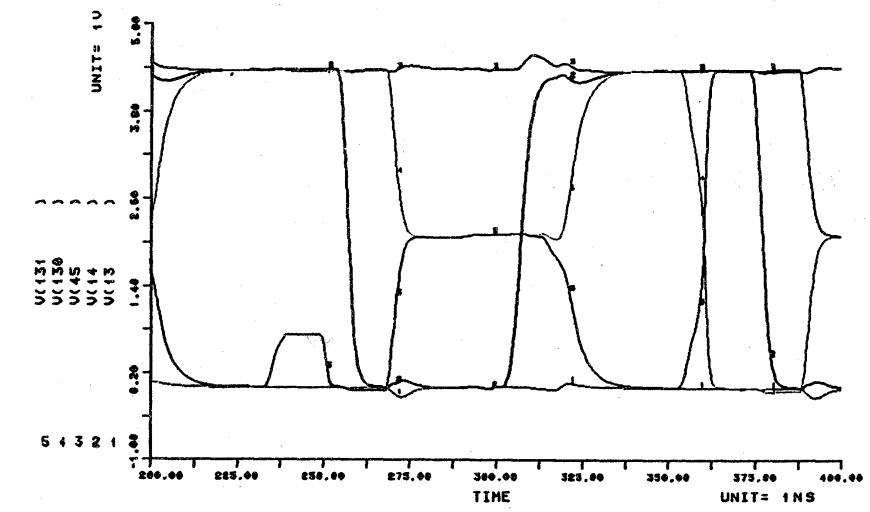


図1.1　回路シミュレーションの概要




```
ENTER COMMAND
G
ENTER COMMAND
```

図1.2　回路シミュレーションの出力結果例

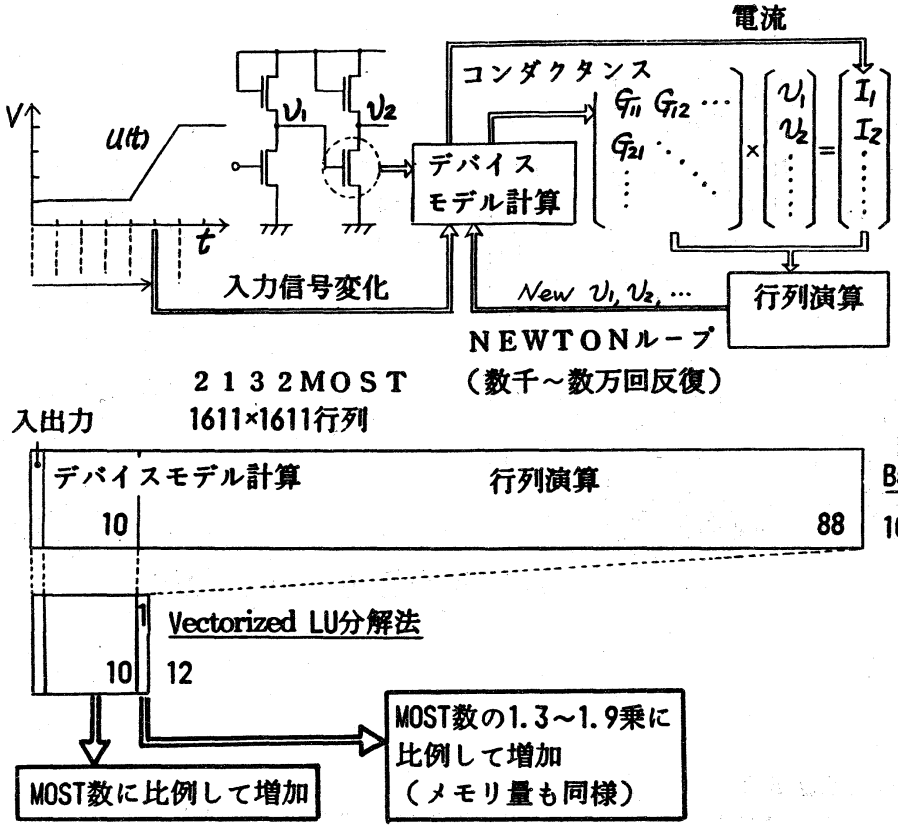


図1.3 大規模回路シミュレーションの性能上の問題点

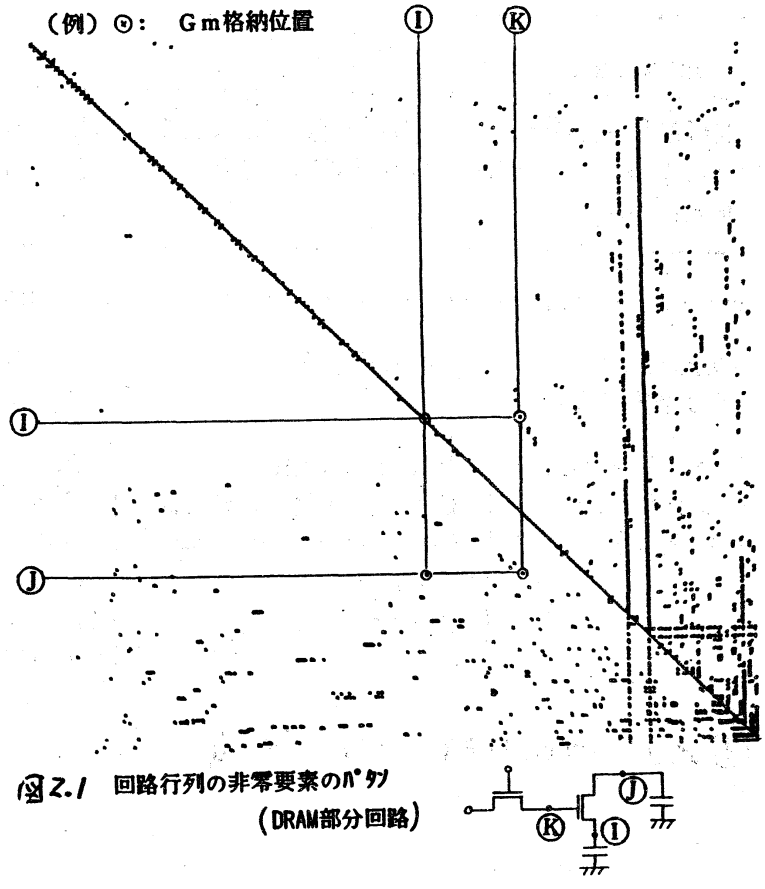
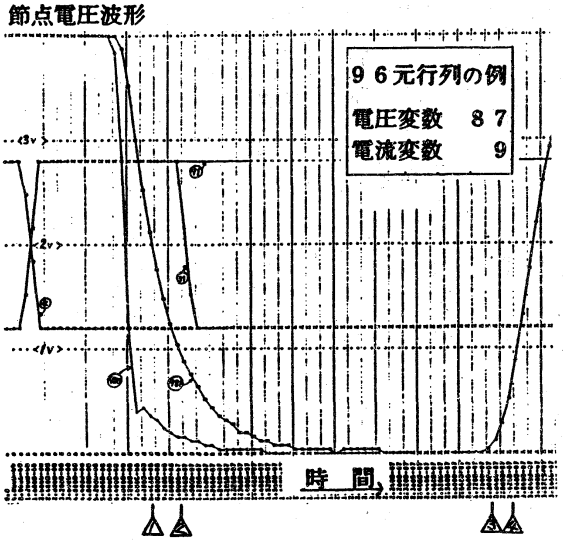


図2.1 回路行列の非零要素のﾊﾟﾀﾝ
（DRAM部分回路）



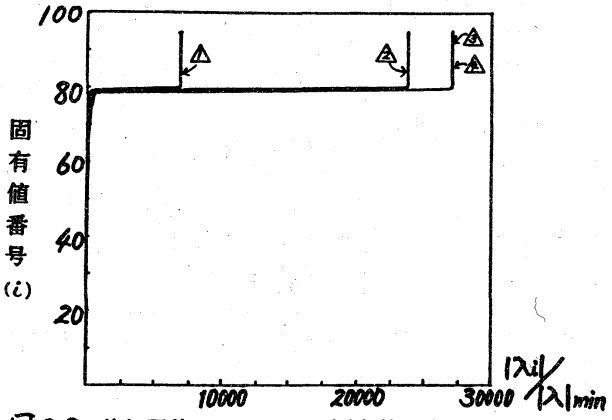


図2.2 前処理前の回路行列の固有値分布

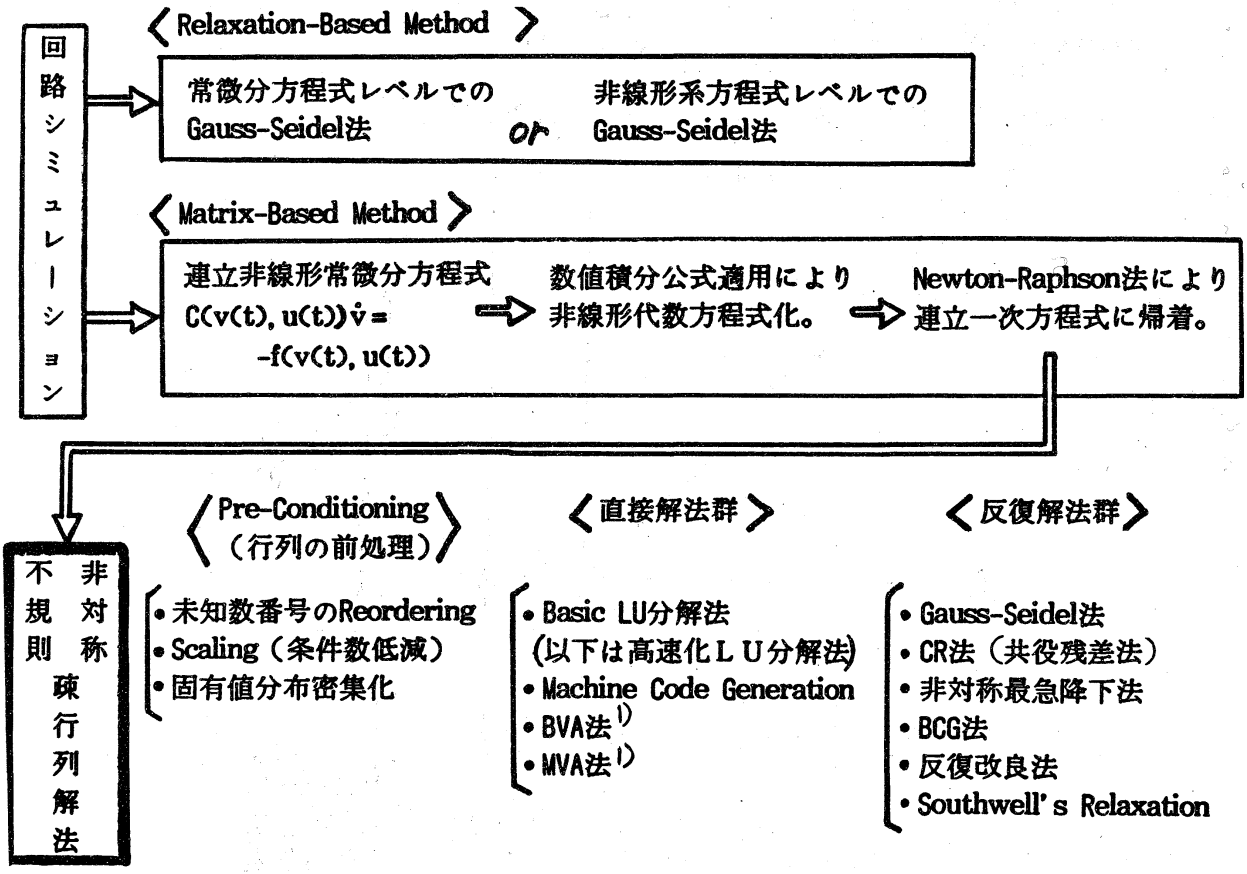


図2.3 回路行列に対する解法

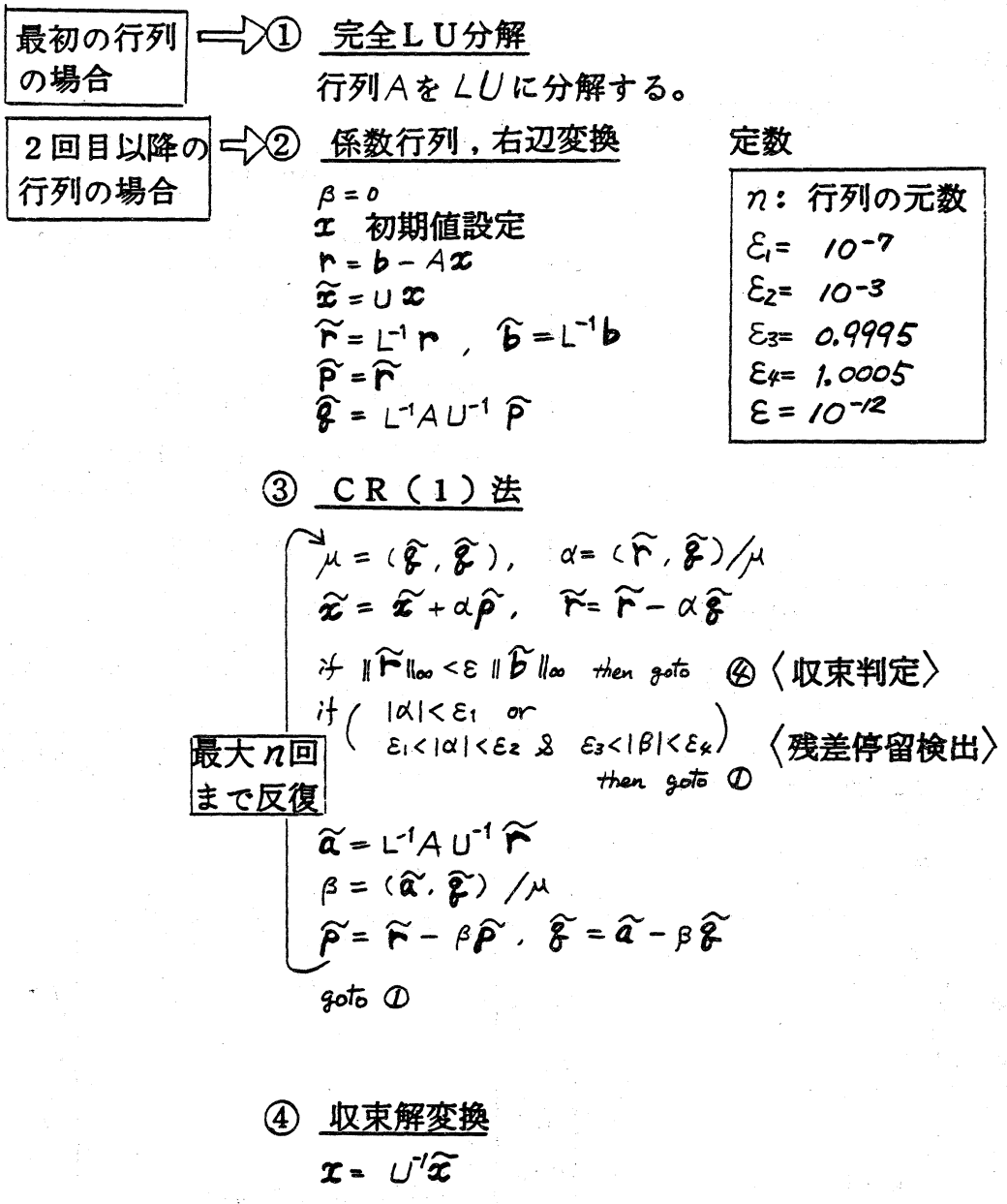


図3.1 完全ＬＵ分解付共役残差法(CLUCR法)の構成

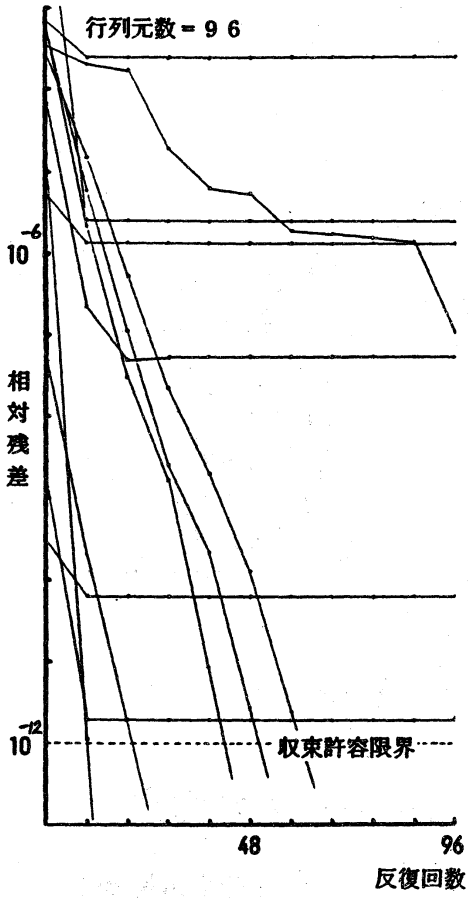


図 4.1 残差減少パタン例

残差停留状態の検出法

（１）残差（最大値ノルム）は一時的に停留することがあり，残差自体の観測では不都合である。

（２）解の修正活動状況（α, β）により検出する。

α ： 最適修正量

β ： 共役直交条件

非収束とみなす条件（次のいずれか）：

① $|\beta|$が１にきわめて近く，$|\alpha|$もあまり大きくない。

⇒ $\beta = (\hat{A}\hat{r}, \hat{A}\hat{p})/(\hat{A}\hat{p}, \hat{A}\hat{p})$

から，$\hat{r} \simeq \hat{p}$ の可能性大。

② $|\alpha|$がきわめて小さい。

⇒ 反復続行しても，改善される可能性が低い。（経験的に）

図 4.2 残差停留状態の検出法

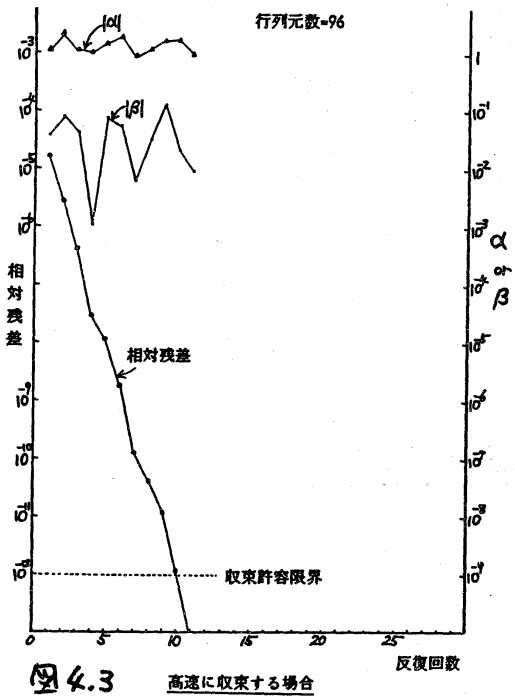


図 4.3 高速に収束する場合

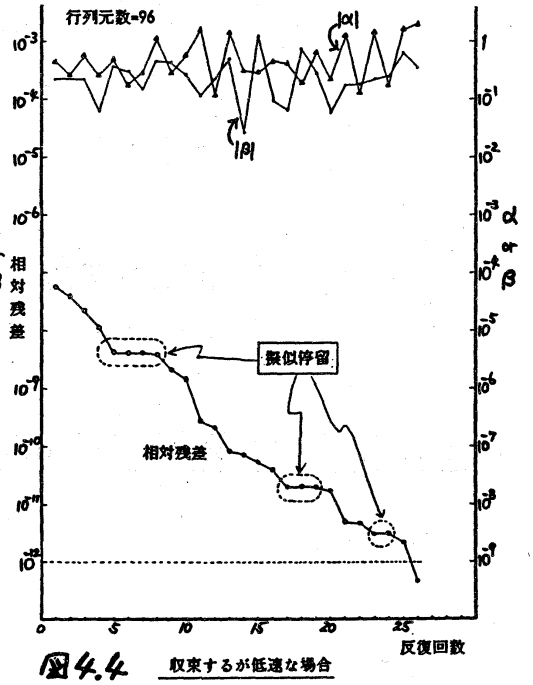


図 4.4 収束するが低速な場合

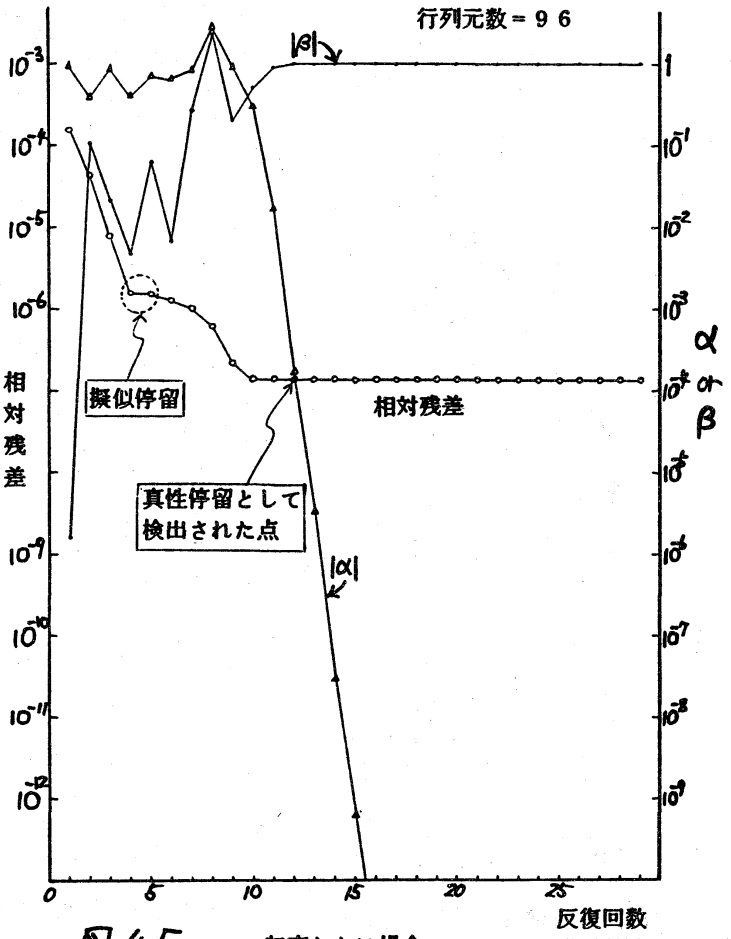


図 4.5　収束しない場合

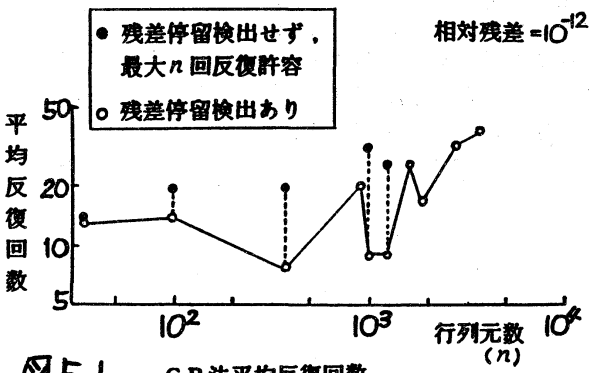


図 5.1　ＣＲ法平均反復回数

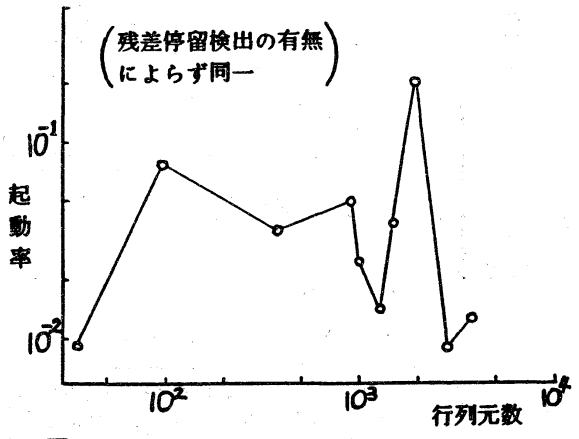


図 5.2　完全ＬＵ分解起動率

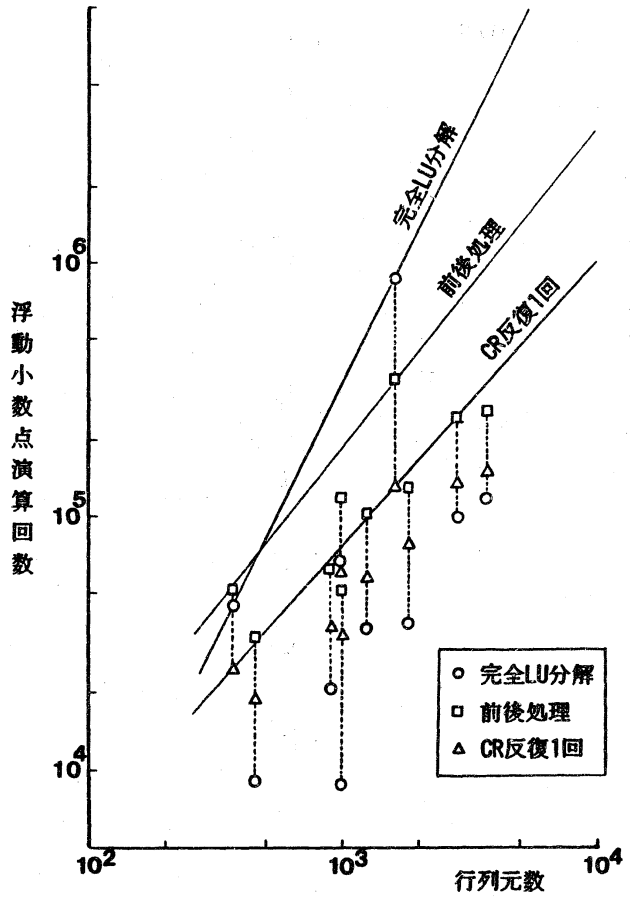


図5.3 CLUCR法の浮動小数点演算の内訳

CLUCR法換算値の条件

	Case-1	Case-2
完全LU分解起動率	0.03	0.05
CR法平均反復回数	10	30
収束許容相対残差	10^{-12}	10^{-12}

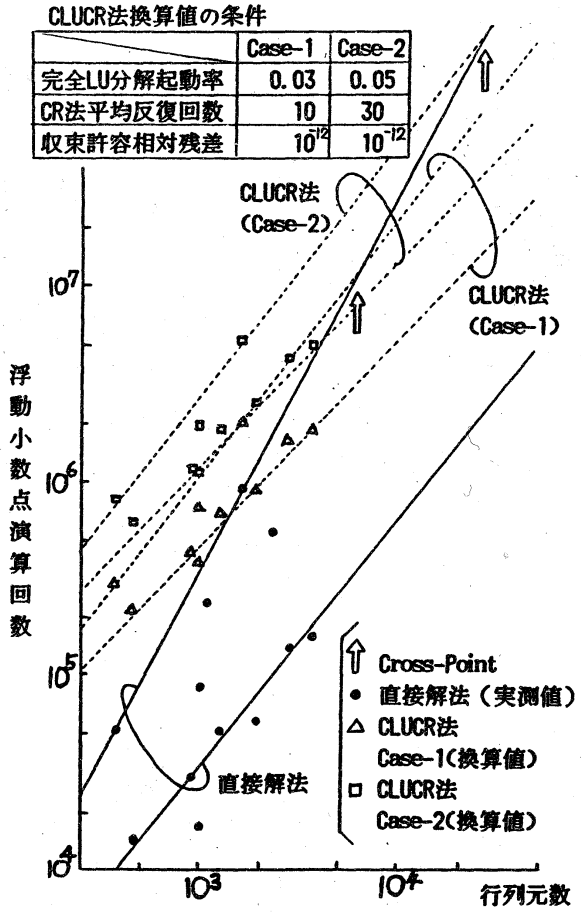


図5.4 行列演算量の比較

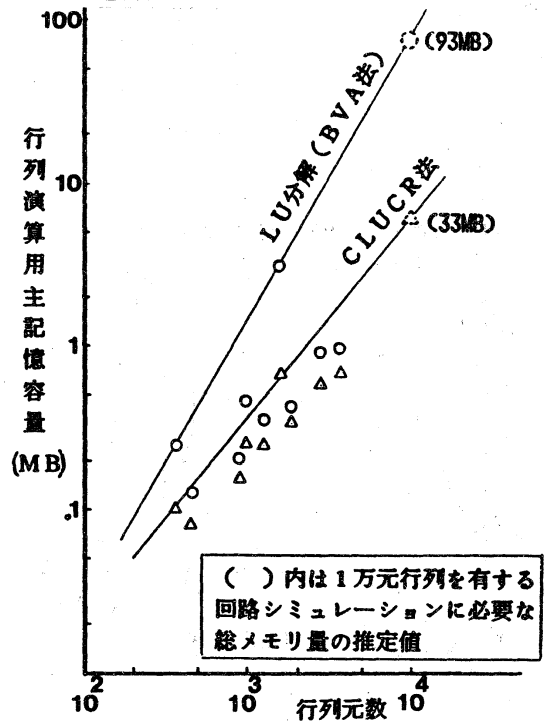


図5.5 行列演算用主記憶容量の比較

注)

CLUCR法ではＬＵ分解起動率が極めて低いことから，それに必要なデータを外部（Ｓ－８１０拡張記憶）に置くものとする。

（注）2サイクル解析。行列は1万元とする。

表5.1 10000 MOST に対する性能推定値(on S810)

	諸元 / 方式	CPU Time	USE Time	主記憶容量	Disk容量	ES[2)]容量	備考
A	ベクトル化方式 LU分解[1)]	5HR	6HR	77MB	—	—	デバイスモデルはスカラ計算
B	ベクトル化方式 LU分解	1.6	1.9	93	—	—	デバイスモデルはベクトル化
C	スカラLU分解 +CLUCR	97	116	33	—	—	〃
D	ベクトル化方式 LU分解+CLUCR (すべて主記憶)	1.3	1.6	98	—	—	〃
E	ベクトル化方式 LU分解+CLUCR (with Disk)	1.3	9.8	34	65MB	—	〃
F	ベクトル化方式 LU分解+CLUCR (with ES)	1.3	1.7	34	—	65MB	〃

1)並列化ブロック分割(BVA)　　2)S810拡張記憶装置

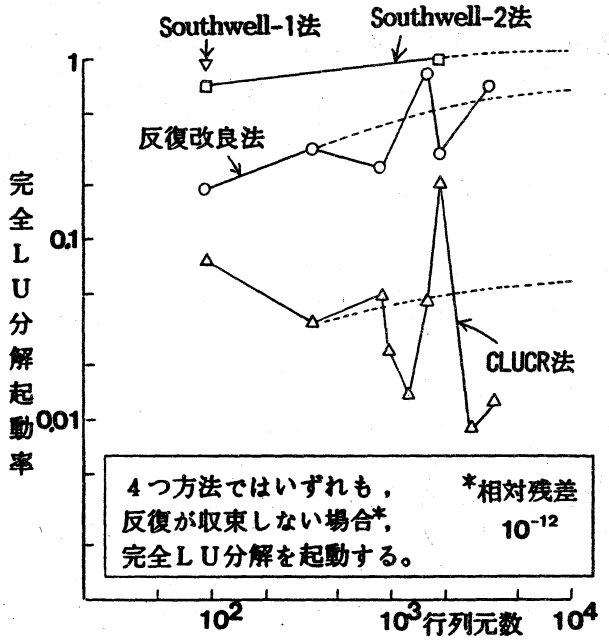


図5.6 他の反復解法との比較